

On the Development of a Framework for Underwater Localization Using All-Source Data

Alberico Menozzi*, Jesse Hansen[†], Gregory Peele*, Stephen Snarski*, Colin Merrick[†]

*Applied Research Associates, Inc., Raleigh, NC, USA – Email: amenozzi@ara.com

[†]Naval Undersea Warfare Center, Newport, RI, USA – Email: jesse.hansen@navy.mil

Abstract—This paper describes the current state of development of a localization framework with the goal of enabling an autonomous underwater vehicle (AUV) to execute long-range precision-navigation missions where positional resets based on GPS and/or transponders are unfeasible and bathymetric maps of sufficient accuracy are not always available. This framework has three main components: a real-time sonar data processing pipeline providing 3-D point-cloud measurements segmented into objects and underwater terrain; a 3-D synthetic natural environment (SNE) integrating geo-registered data from all available sources; a probabilistic localization method linking the SNE to the sonar processing pipeline and producing a probability map over the space of AUV horizontal positions. The current focus and novel contribution of this work is the specific method of exploitation of FLS 3-D point-cloud and 3-D reference map data, and the design of the corresponding probabilistic measurement model for localization. Initial off-line experiments, using real in-water data for which navigational ground truth is available, show the performance of this measurement model and the promise of the overall approach toward achieving horizontal position resets with a 95% circular error probability (CEP95) of less than 20 m.

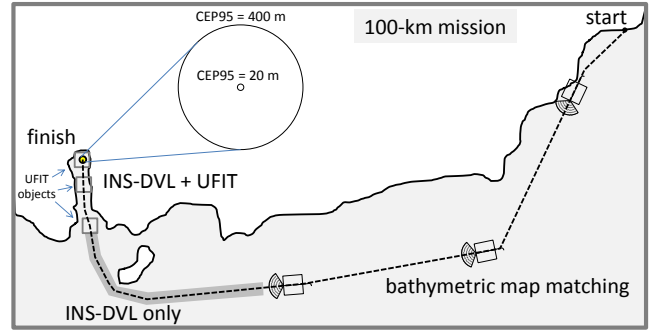


Fig. 1. Current INS-DVL technology accrues an error of about 400 m (CEP95) over a 100-km mission. In the UFIT concept of operation, bathymetric map matching is used in areas where it is feasible, and UFIT is employed in areas where bathymetric maps do not exist but suitable geo-registered reference objects can be inferred from other sources. INS-DVL may be the only choice in transitioning between the two areas. UFIT's goal is to provide horizontal positional resets accurate to within 20 m (CEP95).

I. INTRODUCTION

This paper addresses the problem of enabling an autonomous underwater vehicle (AUV) to execute long-range (e.g., 100 km) precision-navigation missions where positional resets based on GPS and/or transponders are unfeasible and bathymetric maps of sufficient accuracy are not available. Within this context, the objective is to accurately estimate the AUV's horizontal position (e.g., latitude and longitude) using real-time data from the on-board doppler-velocity-log-aided inertial navigation system (INS-DVL) and the AUV's sonar. In areas where reference bathymetric maps are available, a technique that has been successfully demonstrated for long-range navigation consists of using concurrent mapping and localization methods to reduce the positional error growth rate, while also creating a small local map and finding its match within the reference map [1]. However, to allow precise navigation beyond those areas or to increase the likelihood of performing accurate positional resets within them, AUV localization systems must be capable of integrating additional sources of information.

This effort, called *Undersea Fused Inferred Targeting* (UFIT), has as its goal to demonstrate the feasibility of a non-GPS, non-transponder approach to reset horizontal positional error to a 95% circular error probability (CEP95) of less than 20 m in missions where bathymetric map-matching is

not possible (see Fig. 1). Operational requirements include using available 3-D forward-looking sonar (FLS) hardware and fitting the solution within existing size, weight, and power envelopes of typical medium-size AUV's.

UFIT's framework for underwater localization (see Fig. 2) consists of three main components: a real-time *FLS data processing* pipeline that provides dense measurements and discriminates objects from underwater terrain; a 3-D *synthetic natural environment* (SNE) that integrates geo-registered data from all available sources; a *probabilistic localization* method that links the SNE to the FLS processing pipeline and produces a probability map over the space of AUV horizontal positions.

The FLS data used in this study come from a sensor that generates 3-D information on the scene in front of the AUV. This information is used in a real-time processing pipeline to discriminate objects from terrain in the underwater environment (see Fig. 3). The estimated underwater terrain is represented by horizontal bathymetric grids. However, while gridded bathymetric maps adequately capture most of the underwater environment, many objects cannot be adequately represented in this format. Examples include vertical objects such as bridge pilings or cliffs, complex objects such as ship wrecks, and objects that are detached from the ocean floor. Therefore, additional processing is performed to discriminate bathymetry from objects with significant vertical extent. The overall output of the FLS data processing pipeline is a 3-D

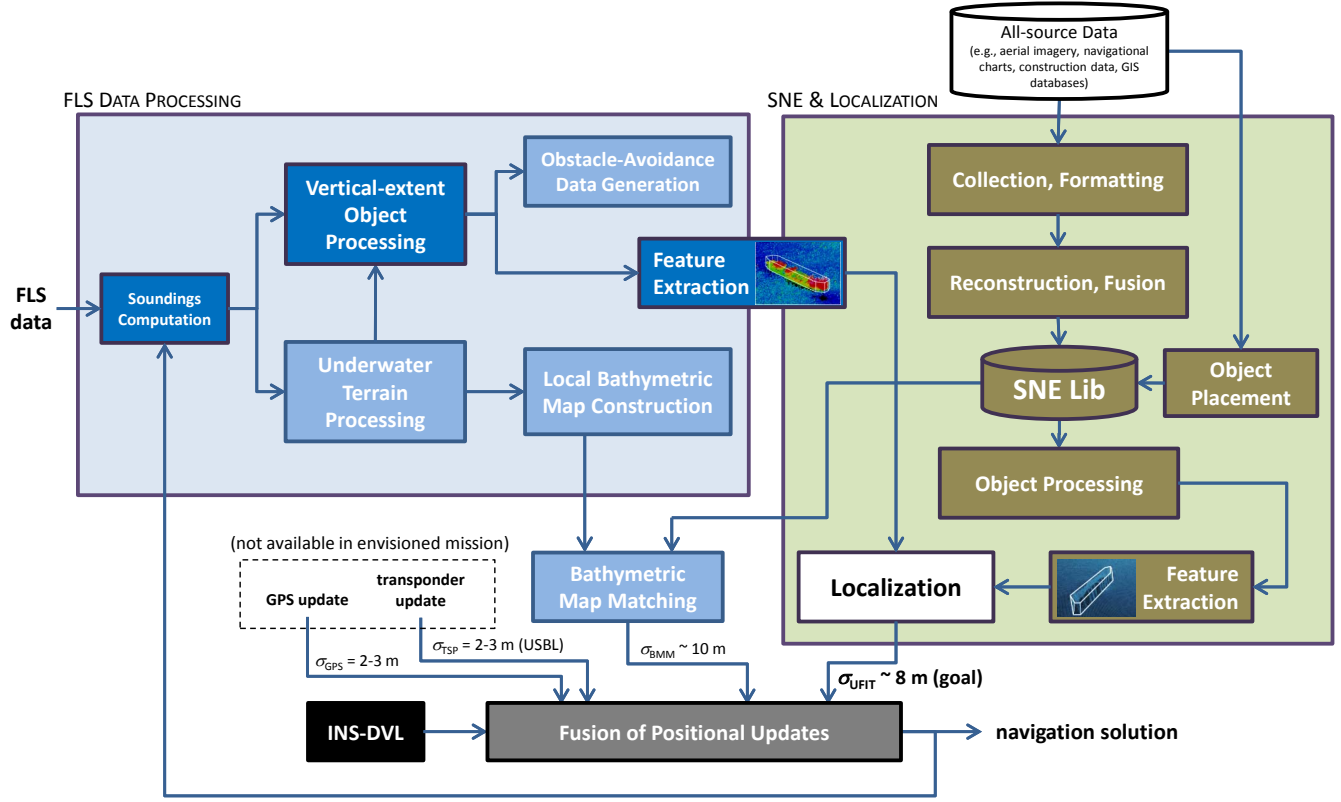


Fig. 2. The main components of the UFIT localization framework are a forward-looking sonar (FLS) data processing pipeline, a 3-D geo-registered synthetic natural environment (SNE), and a localization module that can provide accurate (i.e., CEP95 < 20 m) horizontal positional updates when objects in the SNE are sensed by the FLS. These updates may be fused with other available positional updates to form an integrated navigation solution.

gridded point cloud with segmentation of each point into either underwater *terrain* or vertical-extent *object*.

The main function of the SNE is to implement a geo-registered reference for objects that may be sensed by the FLS during a mission, and therefore provide opportunities for accurate horizontal position resets. Key features of this module include integration of multi-source data (e.g., above-water imagery, construction data, navigation charts) aiding in the inference of the underwater structure of various objects (e.g., bridge pilings, piers, breakwaters), and the ability to add new objects (e.g., pier-side vessels, shipwrecks, dredged channels). Leveraging prior work on the Layered Synthetic Environment Runtime (LaSER) and Layered Terrain Format (LTF) [2], we developed a customized SNE library that includes topography, bathymetry, and a variety of objects of interest. The current implementation supports insertion of bridges, piers, and ships from an existing database, and procedural generation of breakwaters, bridges and piers. The library also provides a set of software services in support of performance-critical operations such as line intersection tests, volumetric collision detection, ray-caster service, and path traversal planning.

The information that is generated from the FLS data processing pipeline and the 3-D SNE module is integrated into a localization algorithm for AUV horizontal position estimation. We are investigating three different approaches to localization

under this framework. The first is based on 3-D object recognition via point-cloud matching. The second is a 2-D shape-matching approach that looks for correlations between a 2-D representation of the SNE and 2-D object polygons produced by FLS processing. The third is a probabilistic localization approach that produces a probability map over the space of possible horizontal positions.

A recent review of the state of the art of AUV localization is given in [3], and includes simultaneous localization and mapping (SLAM) methods using FLS data [4]–[6]. We adopt a probabilistic localization approach, discussed in Sec. II-E, that has gained popularity in AUV navigation state of the art and in the robotics community at large [7]. Within this approach, the current focus and novel contribution of the work presented here is the specific method of exploitation of FLS 3-D point-cloud and 3-D reference map data, and the design of the corresponding probabilistic measurement model.

The long-term goal of this effort is the real-time implementation of the framework in Fig. 2 and the in-water demonstration of the concept of operation illustrated in Fig. 1. The next sections discuss the current state of our method, results based on in-water FLS data, and opportunities and objectives for future work.

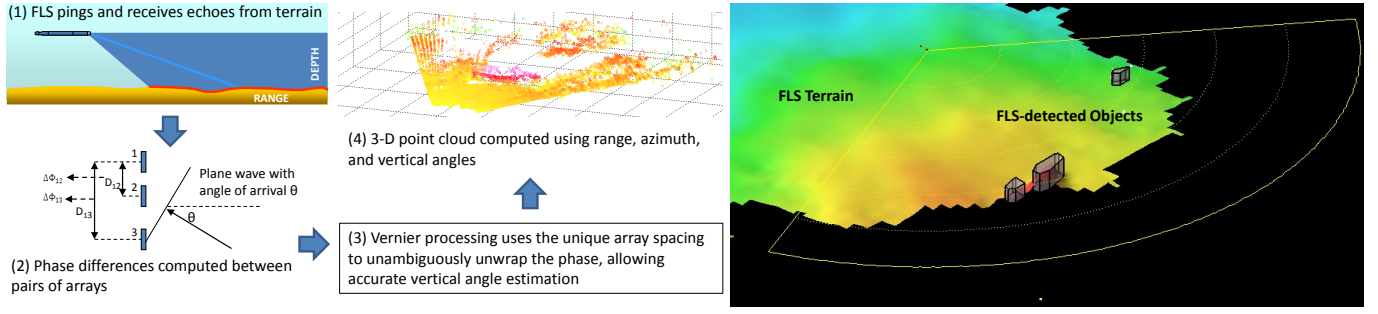


Fig. 3. The left side of the figure shows 3-D sonar processing using beamforming and Vernier interferometry techniques. FLS terrain and vertical-extent object representations are shown on the right side. The overall output of the FLS data processing pipeline is a 3-D gridded point cloud with segmentation of each point into either underwater terrain or vertical-extent object. Each object cluster is also represented by a 2-D polygon with a vertical extent.

II. METHOD

Localization is essentially about finding a good fit between sensor measurement data and a reference map (assumed to be an accurate representation of the operational environment). We have investigated three approaches to this problem. Before describing these approaches, this section discusses the generation of sensor measurement data and the construction of the reference map.

A. FLS Measurement Data

The Naval Undersea Warfare Center (NUWC) Division Newport worked with Teledyne RESON to develop a FLS to meet the following specifications:

- fits within the profile of a 12.75 in-diameter AUV
- 3-D object detection to 300 m range
- high-resolution sector scan imaging to 120 m
- 120-deg lateral field of view
- 3000 m depth capable

The resulting sonar, the RESON 7130 FLS, is a bi-static, dual-frequency system consisting of a transmitter and receiver operating at either 200 kHz or 635 kHz. The receiver contains four horizontal line arrays, three of which are used for reception of 200 kHz signals and one for reception of 635 kHz echoes. All received data are sampled and demodulated in the transducer head. Matched filtering and azimuthal beamforming takes place on a field-programmable gate array (FPGA) in the sonar processing electronics.

The non-uniform vertical spacing of the 200 kHz receive arrays was chosen to enable a technique called Vernier interferometry [8], which uses the unique array spacing to unambiguously unwrap the differential phase, allowing accurate vertical angle estimation without using a fully populated array. Interferometric techniques have been widely studied for application to swath bathymetry and side-scan sonars [9]–[11]. With each vertical angle an empirical measure uncertainty [12] is also computed, and is later used as part of the bathymetric processing.

The range and azimuth/elevation angles from the sonar are used to compute a set of 3-D points in the FLS frame of reference. However, to combine data from multiple pings, these soundings are transformed into a common geo-registered

coordinate system. The on-board INS-DVL provides its estimated attitude (pitch, roll, and azimuth) and horizontal location (latitude and longitude), and a depth estimate is obtained from a pressure sensor. These estimates are combined with the 3-D sonar data to produce a set of geo-registered bathymetric soundings. To combine data across consecutive sonar pings, it is explicitly assumed that navigation error is negligible over short time intervals.

Soundings from the sonar are characterized as either terrain or vertical-extent objects, with terrain represented by a Digital Elevation Model (DEM) with a single depth at each node in a regularly spaced horizontal grid. Note that a DEM is not capable of capturing vertical structures that are characterized by multiple depth values for a given horizontal location. Underwater features such as bridge pilings, wrecks, cliffs, and floating objects fall into this category, and may be considered as obstacles to navigation. Vertical-extent objects are represented by 3-D point clouds, and also parameterized by a 2-D bounding polygon in the horizontal plane and a vertical extent (see right side of Fig. 3). This simplified representation is suitable for both autonomous obstacle avoidance behaviors and mapping of objects for localization.

B. Reference Map

The SNE module provides geo-registered reference map information in real time, while also maintaining off-line usability through 3-D visualization and a graphical user interface. Its run-time representation consists of four conceptual levels: the overall database schema, the instanced models within a database, the layers within each model, and the features within each layer. Features define individual objects within the environment using geometry and other attributes (e.g., material properties). The SNE integrates topography, bathymetry, and aerial/satellite imagery into a single compact data set suitable for both interactive mission planning and AUV localization.

The test environment that was used in this effort is the Narragansett Bay area near Newport, RI. The data set that was used to construct the corresponding SNE consists of navigational chart information, satellite imagery, aerial imagery, topographic data, bathymetric data, and construction data (e.g., bridge and breakwater). Figure 4 shows a 3-D

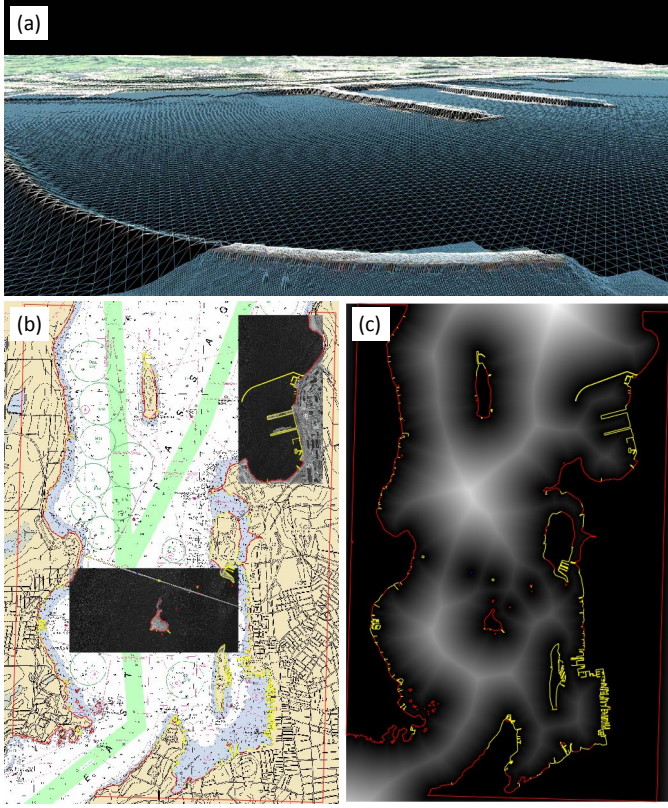


Fig. 4. UFIT's synthetic natural environment (SNE) implements a 3-D map that serves as a reference for underwater localization. It integrates multi-source data with a graphical user interface, visualization, and a set of software services (e.g., ray casting, volumetric collision detection) to support reference-map preparation and real-time localization algorithm computations. Panel (a) shows a 3-D visualization of the SNE; (b) shows a top-down view that highlights the integration of raster navigational charts (RNC) with overhead imagery and electronic navigational chart (ENC) data; (c) shows a top-down view of the ENC data layer with a raster map representation of distances to above-water objects/features.

visualization of the SNE, with a top-down view that highlights the integration of raster navigational charts with overhead imagery and electronic navigational chart data. All these data are geo-registered onto a common 3-D coordinate system. Transformations between different coordinate system representations are also part of the SNE's functionality.

A set of relevant objects within the SNE may be selected for further processing, where attributes are added for use in the localization process. These include geometric properties, material properties, or any attribute that may be needed by a probabilistic model of the FLS sensor's output data. Figure 5 shows examples of aerial photographs that aid in determining attributes of various objects of interest in the operational environment. Attributes that denote the type and underwater slope of a breakwater structure, the "fill factor" of a pier's support structure, and the structure and material of a bridge support pylon may be inferred from these images. Engineering and construction data also provide useful information. (For instance, rubble-mound breakwaters are typically constructed with a slope between 25 and 45 degrees [13].)

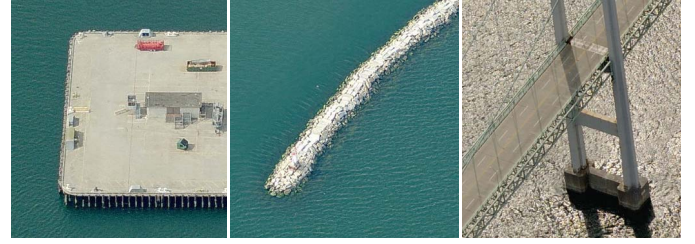


Fig. 5. Aerial photographs aid in determining relevant attributes of various objects of interest in the operational environment. The left panel shows a pier supported by wooden poles with regular spacing. The center panel shows a rubble-mound breakwater. The right panel shows a bridge support pylon as a solid concrete structure. Each of these images contains information that can be used to assign properties to corresponding UFIT SNE objects and infer their underwater geometry.

C. Point-Cloud Matching

The first approach to localization that was investigated is based on 3-D object recognition and 6-DOF pose estimation using point-cloud matching. The idea was to use state-of-the-art point-cloud-based 3-D object recognition approaches [14] to perform localization. Since the FLS produces 3-D gridded point clouds of sensed objects, the approach was to sample UFIT SNE objects into point clouds and look for matches between FLS point clouds and UFIT SNE point clouds. The processing pipeline consisted of transforming UFIT SNE objects to point clouds and computing a measure of similarity between these UFIT SNE point clouds and the point cloud produced by the FLS. The measure of similarity was obtained by first selecting key points from each point cloud and computing a local descriptor for each key point. These descriptors describe the salient characteristics of a local neighborhood of points around the key points. Correspondences were then sought between FLS point cloud descriptors and UFIT SNE point cloud descriptors. The resulting correspondences were grouped and analyzed under certain constraints (i.e., rigid transformation) to ultimately achieve an estimate of a rigid transformation between the FLS point cloud and a UFIT SNE object point cloud. Finding such transformation is equivalent to localizing the vehicle within the SNE.

This point cloud registration approach did not produce desired results because it is too sensitive to noise, and the local features of the FLS occupancy grid did not match well enough with the local features of the corresponding synthetic point cloud (see Fig. 6).

D. 2-D Map Matching

Another localization approach that was investigated consists of matching a local 2-D map based on sensor data with a 2-D reference map based on the SNE. The match is searched over the space of translations of the local map within the reference map (azimuth is assumed to be known with enough accuracy). This 2-D reference map m_{2D} is a sea-level cross section of the SNE that contains geo-registered features such as the coastline, man-made shoreline improvements (breakwater, piers, jetties), bridge pilings, and other features. The FLS data are used to

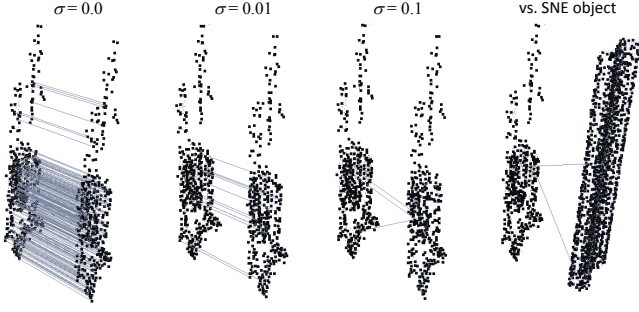


Fig. 6. Four different point-cloud pairs, where correspondences between key-point descriptors are shown as straight lines. The first pair from the left consists of two identical copies of FLS point-cloud data from imaging a bridge pylon, in which case each descriptor is perfectly matched. The second and third pairs from the left each consist of the original point cloud and a slightly perturbed version of itself, resulting in very rapid degradation of correspondences. The last pair consists of the original point cloud and one resulting from sampling the corresponding bridge pylon object in the SNE, in which case no valid correspondences are found.

construct a local map m_{local} defining unoccupied, occupied, and unknown regions. Occupied regions are defined by 2-D polygons that bound the horizontal extent of objects detected by the FLS (see Sec. II-A and Fig. 3).

Both m_{2D} and m_{local} are converted from vector/polygon representations to 2-D gridded occupancy maps with adjustable grid resolution. The gridded m_{2D} is used to define two raster maps of distance measures: distances D_0 to unoccupied cells and distances D_1 to occupied cells. D_0 represents the distance from any grid cell in m_{2D} to the nearest unoccupied cell, while D_1 represents the distance from any grid cell in m_{2D} to the nearest occupied cell. Both D_0 and D_1 are computed offline before the start of a mission.

During a mission, the AUV's local occupancy map m_{local} is used to localize within the global map m_{2D} . For each hypothesized translation of m_{local} , two distance measures are computed. The first is the total distance of every occupied grid cell in m_{local} from the nearest object in m_{2D} :

$$d_1(\Delta x, \Delta y) = \sum_{x,y} f_1(x - \Delta x, y - \Delta y) D_1(x, y),$$

where f_1 takes a value of 1 for all occupied cells and zero otherwise. The second is the total distance of every unoccupied grid cell in m_{local} from the nearest unoccupied cell in m_{2D} :

$$d_0(\Delta x, \Delta y) = \sum_{x,y} f_0(x - \Delta x, y - \Delta y) D_0(x, y),$$

where $f_0(x, y)$ takes a value of 1 for all unoccupied cells and zero otherwise. These distances are combined into a single distance measure

$$d(\Delta x, \Delta y) = \frac{d_1(\Delta x, \Delta y) + d_0(\Delta x, \Delta y)}{n_1 + n_0}, \quad (1)$$

where n_1 and n_0 are the number of occupied and unoccupied grid cells, respectively. The horizontal translation $(\Delta x^*, \Delta y^*)$ of the local map that minimizes (1) is chosen for use in localization: $(\Delta x^*, \Delta y^*) = \arg \min_{\Delta x, \Delta y} d(\Delta x, \Delta y)$. This

result can be further refined by interpolating around the location of the minimum distance via a 2-D polynomial fit. Some care must be taken to avoid degenerate cases, such as having no or few occupied regions in m_{local} .

This method is subject to two potential issues. The first is tied to the generation of 2-D object polygons in m_{local} . The FLS data processing pipeline does not produce 2-D object polygons at sea level. It instead returns a 2-D polygon encompassing a horizontal cross section of sensed object data closest to the surface. In some situations (e.g., a sloped breakwater) this object polygon may be larger or different in shape than the corresponding object polygon in the reference map m_{2D} , which leads to problems in map matching. The second issue is that this method does not take into account past information, and may jump between alternate hypotheses when objects with similar 2-D geometries are in m_{2D} .

Both issues may be overcome by integrating this method into the probabilistic localization approach described in the next subsection, where it would contribute a computationally efficient input to the probabilistic measurement model.

E. Probabilistic Localization

The object of localization is to estimate the AUV's state \mathbf{x}_t at time t given a map m of the environment and other available information (e.g., sensor data). The probabilistic framework that is commonly used as the basis for solving this problem is the *Bayes filter*, which produces a belief distribution $\text{bel}(\mathbf{x}_t|m)$ over the state space (i.e., it assigns a probability to each possible state) in the following recursive manner:

$$\overline{\text{bel}}(\mathbf{x}_t|m) = \int p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t}, \Delta \mathbf{x}_t, m) \text{bel}(\mathbf{x}_{t-\Delta t}|m) d\mathbf{x}_{t-\Delta t}$$

$$\text{bel}(\mathbf{x}_t|m) = \eta p(Z_t|\mathbf{x}_t, m) \overline{\text{bel}}(\mathbf{x}_t|m),$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-\Delta t}, \Delta \mathbf{x}_t, m)$ is a probabilistic motion model, $p(Z_t|\mathbf{x}_t, m)$ is a probabilistic measurement model, and η is a normalization constant. The motion model expresses the probability of \mathbf{x}_t conditioned on the previous state $\mathbf{x}_{t-\Delta t}$, motion data $\Delta \mathbf{x}_t$, and the map m . The measurement model expresses the probability of a set of sensor measurements Z_t , conditioned on the state and the map. In this effort, we use *Monte Carlo Localization* (MCL), which is an implementation of the Bayes filter that uses a *particle filter*, thereby representing the belief distribution by a set $\mathcal{X}_t = \{\mathbf{x}_t^{[1]}, \mathbf{x}_t^{[2]}, \dots, \mathbf{x}_t^{[n_k]}\}$ of n_k state samples (i.e., particles) drawn from it [7]. The denser are the samples in a region of the state space, the higher is the probability that the AUV is in that region.

The MCL algorithm used in this effort is shown in Table I. It implements a single step of the Bayes filter recursion by taking an existing set of particles $\mathcal{X}_{t-\Delta t}$, a set of sensor measurements Z_t , and the AUV's own motion estimate $\Delta \hat{\mathbf{x}}_t \approx \mathbf{x}_t - \mathbf{x}_{t-\Delta t}$ (assumed to be very accurate for small Δt), and returning an updated set of particles. First, each particle in the set is moved according to the probabilistic motion model. This is implemented in lines 4 and 5 of the algorithm, where Q is a noise covariance matrix. If a particle falls on a map object as

TABLE I
MONTE CARLO LOCALIZATION (MCL) ALGORITHM

Algorithm MCL($\mathcal{X}_{t-\Delta t}$, Z_t , $\Delta \hat{\mathbf{x}}_t$, rspl_prev , rspl_now)

```

1: static  $\mathbf{Q}$ ,  $m$ ,  $\mathbf{D}_m$ ,  $\alpha$ ,  $\beta$ 
2:  $\bar{\mathcal{X}}_t \leftarrow \emptyset$ ,  $n_k \leftarrow |\mathcal{X}_{t-\Delta t}|$ 
3: for  $k = 1$  to  $n_k$  do
4:    $\epsilon_t^{[k]} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ 
5:    $\mathbf{x}_t^{[k]} \leftarrow \mathbf{x}_{t-\Delta t}^{[k]} + \Delta \hat{\mathbf{x}}_t + \epsilon_t^{[k]}$ 
6:   if  $\mathbf{D}_m(\mathbf{x}_t^{[k]}, y_t^{[k]}) < \beta$  then {particle too close to a map object}
7:      $p \leftarrow 0$ 
8:   else
9:      $p \leftarrow p(Z_t | \mathbf{x}_t^{[k]}, m)$ 
10:  end if
11:  if  $\text{rspl\_prev}$  then
12:     $w_t^{[k]} \leftarrow 1$ 
13:  end if
14:   $w_t^{[k]} \leftarrow p^\alpha w_t^{[k]}$ 
15:   $\bar{\mathcal{X}}_t \leftarrow \bar{\mathcal{X}}_t + (\mathbf{x}_t^{[k]}, w_t^{[k]})$ 
16: end for
17: if  $\text{rspl\_now}$  then
18:    $\mathcal{X}_t \leftarrow \emptyset$ ,  $n_k \leftarrow |\bar{\mathcal{X}}_t|$ 
19:   for  $k = 1$  to  $n_k$  do
20:     draw  $j \in \{1, \dots, n_k\}$  with probability  $\propto w_t^{[j]}$ 
21:      $\mathcal{X}_t \leftarrow \mathcal{X}_t + (\mathbf{x}_t^{[j]}, \frac{1}{n_k})$ 
22:   end for
23:   return  $\mathcal{X}_t$ 
24: else
25:   return  $\bar{\mathcal{X}}_t$ 
26: end if

```

a result of the motion, then its corresponding measurement probability is set to zero (lines 6 and 7). Otherwise, it is computed using the probabilistic measurement model (line 9). The rest of the algorithm proceeds with the standard steps of using the measurement probability to update the importance weights and performing resampling—implemented with a low-variance sampler—if requested via the flag `rspl_now`. After resampling, the particles are distributed approximately according to $\text{bel}(\mathbf{x}_t | m)$. The development and implementation of the probabilistic measurement model in line 9 is the most critical piece of MCL and the rest of this section is dedicated to it.

F. Computation of $p(Z_t | \mathbf{x}_t, m)$

The main idea for computing the probabilistic measurement model is to represent both 3-D FLS point-cloud data and the 3-D SNE by a number n_s of 2-D subsets and then compute the likelihood of each 2-D point-cloud subset Z_t^i conditioned on the corresponding 2-D SNE subset m^i . The overall measurement likelihood is then computed as the product of the individual 2-D subset likelihoods:

$$p(Z_t | \mathbf{x}_t, m) = \prod_{i=1}^{n_s} p(Z_t^i | \mathbf{x}_t, m^i). \quad (2)$$

This approach has two main benefits. First, it drastically reduces the computational complexity compared to performing the same task with the full 3-D data set. Second, it makes it easier to design and implement probabilistic models. Judicious choice of 2-D subsets is key to taking advantage of these benefits while still preserving important information contained in the 3-D data and in the reference map. Our choice is to

sample 3-D information via n_s vertical 2-D cutting planes fanning out from the estimated/hypothesized FLS location.

The 2-D measurement data subsets (or “data slices”) are generated by sampling the 3-D gridded point cloud G_t produced by the FLS at time t with n_s vertical cutting planes that are specified with respect to the AUV’s estimated state $\hat{\mathbf{x}}_t$. Points closer than a threshold distance to the i -th plane are projected onto it, and these projections constitute a data slice Z_t^i . The vertical cutting planes are oriented with an azimuth of

$$\psi_Z^i = \hat{\psi} - \frac{\Delta\psi}{2} (n_s - 2i + 1),$$

where $\hat{\psi}$ is the AUV’s estimated azimuth. They extend from the FLS estimated location (\hat{x}, \hat{y}) to a maximum range r_{\max} and from zero depth to the maximum depth found in the data for that slice (see Fig. 7). The set $Z_t = \{Z_t^i : i = 1, 2, \dots, n_s\}$ of data slices constitutes the measurement data at time t .

The 2-D map subsets (or “map slices”) are generated by sampling the 3-D SNE with n_s vertical cutting planes that are specified with respect to the state \mathbf{x}_t . In the MCL algorithm, the k -th particle generates a set of cutting planes with respect to its own state $\mathbf{x}_t^{[k]}$. The vertical cutting planes are oriented with an azimuth of

$$\psi_m^i = \psi^{[k]} - \frac{\Delta\psi}{2} (n_s - 2i + 1),$$

where $\psi^{[k]}$ is the particle’s azimuth. They extend from the particle’s location $(x^{[k]}, y^{[k]})$ to a maximum range r_{\max} and from zero depth to a maximum depth d_{\max}^i (see Fig. 8). The set $\{m^i : i = 1, 2, \dots, n_s\}$ of map slices constitutes the map data for that particle.

Because it is done by each of a potentially large number of particles, the process of sampling the map m is the main draw of computing resources. It is critical to real-time operation to reduce this cost as much as possible. In this approach, instead of executing expensive ray-casting directly in the 3-D SNE, a 2-D depth raster map \mathbf{D}_m is generated a-priori from the SNE and then sampled using *ray marching* (based on *Bresenham* line rendering [15]). (An illustration of this is shown on the left side of Fig. 10.) Minimal processing is then performed on the results to generate a map slice m^i (e.g., filling in gaps due to sampling near-vertical map objects). Each map slice is then used to construct the corresponding $p(r, d | \mathbf{x}_t, m^i)$, which models the probability of getting a FLS data point at range r and depth d , given \mathbf{x}_t as the current state and m^i as the current map slice. This model can be written as

$$p(r, d | \mathbf{x}_t, m^i) = p(r | d, \mathbf{x}_t, m^i) p(d | \mathbf{x}_t, m^i), \quad (3)$$

and its sample space is defined to be $\Omega^i = [0, r_{\text{out}}] \times [0, d_{\max}^i]$, where r_{out} is the range just beyond the maximum sensor range (i.e., $r_{\text{out}} = \min\{r : r > r_{\max}\}$), and d_{\max}^i is the maximum depth established for map slice m^i . The reason for r_{out} is to include in the sample space the *absence* of a FLS data point. This event is handled, for analytical purposes, by inserting an artificial “no-data” point just beyond the reach of the sensor (i.e., at r_{out}).

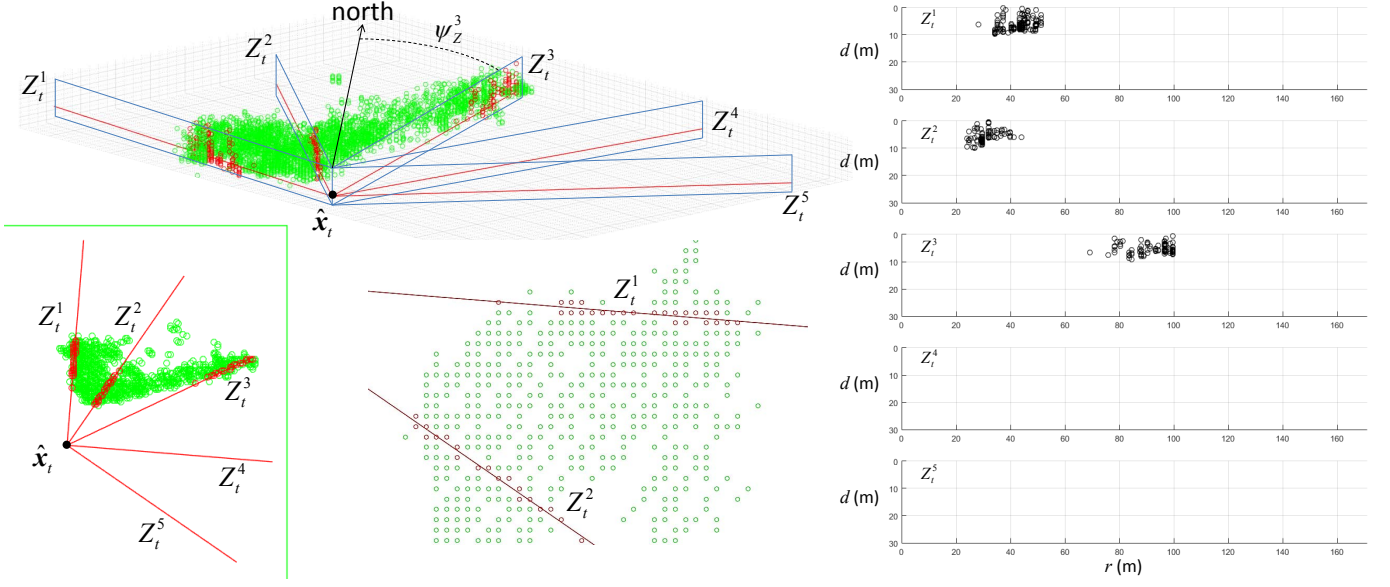


Fig. 7. The 3-D gridded point cloud G_t generated by the FLS at time t is sampled using a set of n_s vertical cutting planes that are specified with respect to the AUV's estimated state $\hat{\mathbf{x}}_t$. Points closer than a threshold distance to each plane are projected onto it, and these projections constitute a data slice Z_t^i . The set $Z_t = \{Z_t^i : i = 1, 2, \dots, n_s\}$ of data slices constitutes the measurement data at time t . The left side of this figure shows 3-D and top-down views of this process. The right side shows corresponding data slices expressed as values of depth d and range r from the FLS.

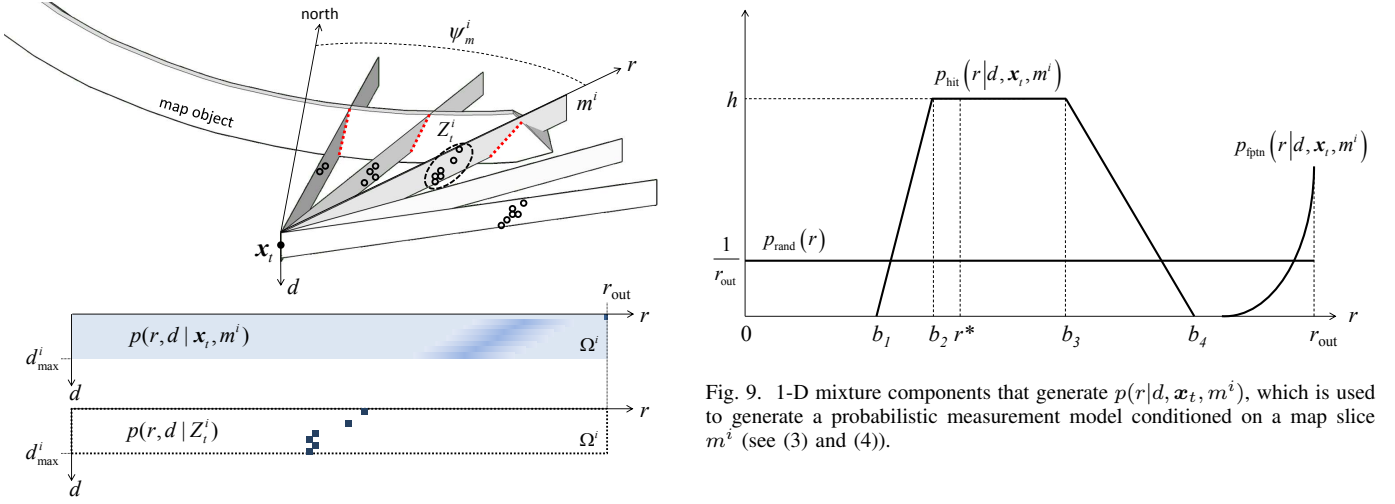


Fig. 8. The 3-D SNE is sampled using a set of n_s vertical cutting planes that are specified with respect to the state \mathbf{x}_t in the same way that the data cutting planes are specified with respect to the AUV's estimated state $\hat{\mathbf{x}}_t$ (see Fig. 7). Each map slice m^i is used to generate a probabilistic measurement model conditioned on it, which is then used to evaluate the likelihood of the corresponding measurement data slice Z_t^i (see (5)).

Construction of $p(r, d | \mathbf{x}_t, m^i)$ proceeds by first assuming that $p(d | \mathbf{x}_t, m^i) = 1/d_{\max}$, meaning that all depths in $[0, d_{\max}^i]$ are considered equally likely to have a data or “no-data” point. The probability of getting a data point at a range r , given a depth d along with \mathbf{x}_t and m^i , is then built as a mixture of three distributions (see Fig. 9):

$$p(r | d, \mathbf{x}_t, m^i) = \frac{\phi_{\text{rand}} p_{\text{rand}} + \phi_{\text{hit}} p_{\text{hit}} + \phi_{\text{fptn}} p_{\text{fptn}}}{\phi_{\text{rand}} + \phi_{\text{hit}} + \phi_{\text{fptn}}}, \quad (4)$$

Fig. 9. 1-D mixture components that generate $p(r | d, \mathbf{x}_t, m^i)$, which is used to generate a probabilistic measurement model conditioned on a map slice m^i (see (3) and (4)).

where $p_{\text{rand}} = \frac{1}{r_{\text{out}}}$ models totally random data, $p_{\text{fptn}} = \eta_{\text{fptn}} e^{-\gamma(r_{\text{out}} - r)}$ models the union of false positives and true negatives, $p_{\text{hit}} = \eta_{\text{hit}} \text{trap}(r; r^*, h, b_1, b_2, b_3, b_4)$ is a trapezoidal density function that models the probability of getting a data point when there is a map object at range r^* , and ϕ_{rand} , ϕ_{hit} , and ϕ_{fptn} are mixture coefficients. All mixture distributions are zero outside of $[0, r_{\text{out}}]$ and are made to integrate to one via normalization constants (i.e., η_{fptn} and η_{hit}). The parameter γ of p_{fptn} is a decreasing function of d . The parameters of the trapezoidal density function in p_{hit} are a function of both depth and the attributes of the SNE object being sampled in m^i (see Fig. 5), and ϕ_{hit} is a decreasing function of r^* . Figure 10 shows an illustration of the implementation of (3) for the case of a rubble-mound breakwater.

To compute $p(Z_t^i | \mathbf{x}_t, m^i)$, the distribution of data samples

in Z_t^i and the probabilistic model $p(r, d|\mathbf{x}_t, m^i)$ of (3) are compared as follows:

$$p(Z_t^i|\mathbf{x}_t, m^i) = \int_{(r,d) \in \Omega^i} \sqrt{p(r, d|Z_t^i)p(r, d|\mathbf{x}_t, m^i)}, \quad (5)$$

where $p(r, d|Z_t^i)$ is a probability density function obtained directly from the data samples in Z_t^i . The right-hand side of (5) is the Bhattacharyya coefficient [16], which is in $[0, 1]$ and is an approximate measure of the amount of overlap between two distributions.

Finally, substituting (5) in (2) gives the measurement probability that is needed in line 9 of the MCL algorithm of Table I.

III. RESULTS

This section presents experimental results based on using a Matlab implementation of the algorithm described in Sec. II-E to process real FLS data for which navigational ground truth is available.

The test vehicle that was used to collect the data is the widely used 12.75 in-diameter AUV, such as the Remote Environmental Measuring UnitS (REMUS) 600 (Kongsberg-Hydrod) or the Bluefin 12D (Bluefin Robotics), equipped with the FLS described in Sec. II-A. Two in-water runs were performed. The first was an approach and swim along a pier, and the second was a straight run by a rubble-mound breakwater. During each run, the true location of the AUV was logged with an accuracy of about 0.5 m. The FLS data processing pipeline was executed on-board at a rate of 1 Hz, and the corresponding set $\{G_t : t = 0, 1, \dots, t_{\text{end}}\}$ of 3-D gridded point clouds was logged as measurement data for off-line processing. The INS-DVL state estimate $\hat{\mathbf{x}}_t$ was also logged at the same rate, to be used off line in the computation of $\Delta\hat{\mathbf{x}}_t$.

Each G_t is defined along the horizontal Universal Transverse Mercator (UTM) easting and northing directions and the down direction, and bounded by a bounding box centered at the FLS's location. The grid resolution is 1 m along easting and northing and 0.25 m along the down direction. The state of the vehicle is defined to be $\mathbf{x} = [x \ y \ \psi]^T$, where x and y are easting and northing coordinates and ψ is azimuth. Depth, roll, and pitch are considered to be estimated by the on-board INS-DVL with enough accuracy that they are not part of the localization states.

In any given MCL experiment, the initial set \mathcal{X}_0 is composed of n_k particles with horizontal locations sampled from a normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where $\boldsymbol{\mu} \sim \mathcal{N}((x_0, y_0), \Sigma)$, (x_0, y_0) is the true initial location of the AUV, and $\Sigma = \text{diag}([(400/2.4477)^2, (400/2.4477)^2])$ (i.e., equivalent to 400 m CEP95). The initial azimuths are sampled from $\mathcal{N}(\psi_0, (\frac{1\pi}{180})^2)$, where ψ_0 is the true initial azimuth of the AUV. The rest of the constant parameters of the probabilistic localization algorithm are as follows: $Q = \text{diag}([0.1^2, 0.1^2, (\frac{0.5\pi}{180})^2])$, $\phi_{\text{rand}} = 3$, $\phi_{\text{fptn}} = 0.4$, $\alpha = 0.75$, $\beta = 15$, $\Delta t = 1$, and a particle resampling interval of $5\Delta t$.

Representative results are shown in Fig. 11, where each panel shows the outcome of $N = 50$ separate localization

experiments with n_k particles. For each experiment there is an 'x' marking the center μ of the initial particle distribution, and a corresponding 'o' marking the square meter with highest density of particles after some time, representing the most likely location of the AUV at that time. The true horizontal trajectory of the AUV is shown as a solid line. To enhance legibility, number labels are placed next to 'o' marks denoting the quantity of those marks in that neighborhood. The number in bold font denotes the quantity of 'o' marks that are within 20 m of the true AUV location.

The stated goal is to demonstrate that, for N large-enough, at least 95% of the 'o' marks fall within 20 m of the true AUV location. The results in Fig. 11 show that achieving this goal with MCL is not just a matter of increasing the number of particles. A large number of particles does help in discriminating among multiple hypotheses with small but discernible differences between them. For example, in the pier experiment (top row of Fig. 11) increasing the number of particles helps in reducing outliers and false positives and in discriminating the right pier from a very similar one to its north (pier width being the only difference). On the other hand, in the breakwater experiment (bottom row of Fig. 11) there are several hypothetical AUV trajectories, slightly translated from the true one, that result in the same match to the sensor data as the true trajectory. This is reflected in the distribution of 'o' marks for both the $n_k = 5000$ and $n_k = 10000$ cases. As the AUV swims past the breakwater, however, the sensed data allow the MCL algorithm to discriminate among the multiple hypotheses that it carried up to that point (see Fig. 12).

Particle management, vehicle trajectory, and the characteristics of the operating environment all play a role in the outcome of MCL. At the core of localization performance in this specific application is the probabilistic measurement model being developed under this effort and described in this paper, and the corresponding exploitation of FLS 3-D point-cloud and 3-D reference map data.

IV. FUTURE WORK

Besides solidifying localization performance, the long-term goal of this effort is real-time implementation and the in-water demonstration.

Computational effort is directly proportional to the number of particles n_k . In our off-line experiments, on average, real-time performance was achieved when running 500 particles at 13% usage of a Intel Core i7-4900MQ CPU clocked at 2.8 GHz. Based on current results, at least a $10\times$ improvement is needed to achieve good real-time localization performance. (Making full use of the existing CPU would itself yield about a $7\times$ improvement.) To this end, we plan to work on transitioning the existing MCL Matlab-based code to C/C++ and take advantage of graphics libraries and GPU hardware to implement the 2-D processing parts of the algorithm.

Another area of future work, which would likely lead to a decrease in the number of particles needed, is the implementation of relevant state-of-the-art advances in particle filtering

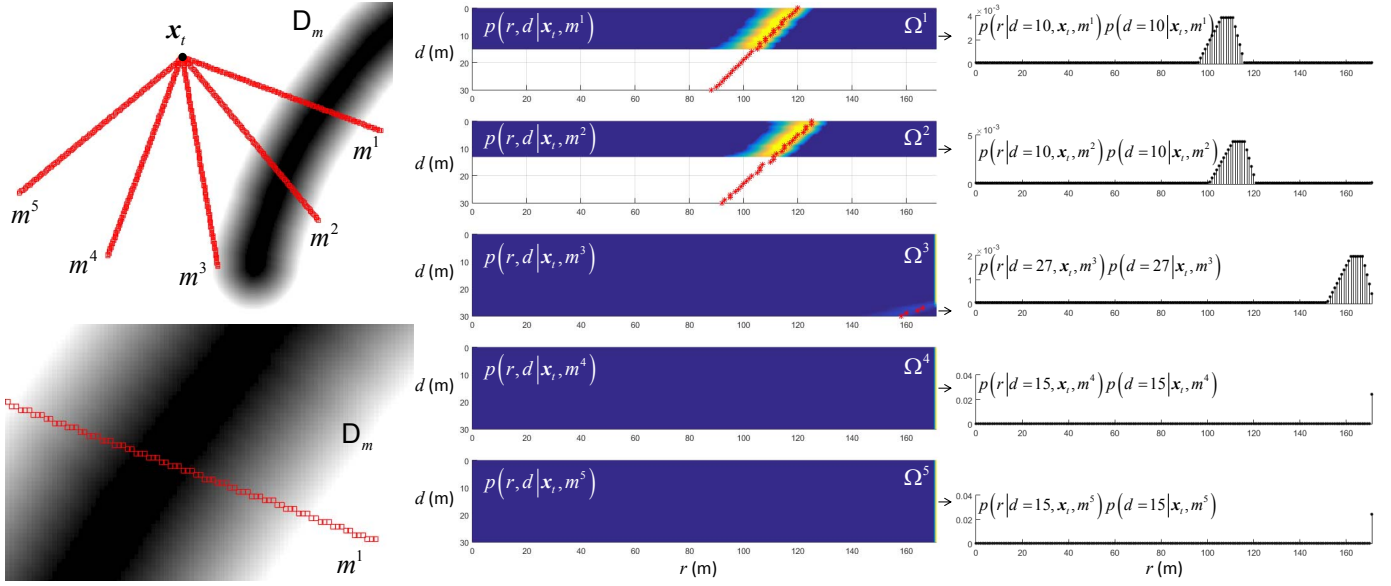


Fig. 10. Instead of executing expensive ray-casting directly in the 3-D SNE, a 2-D depth raster map D_m is generated a-priori from the SNE and then sampled using *ray marching* (based on *Bresenham* line rendering [15]). The resulting map slices are used to generate corresponding probabilistic measurement models $p(r, d | x_t, m^i)$. (See (3) and (4), and Fig. 9.)

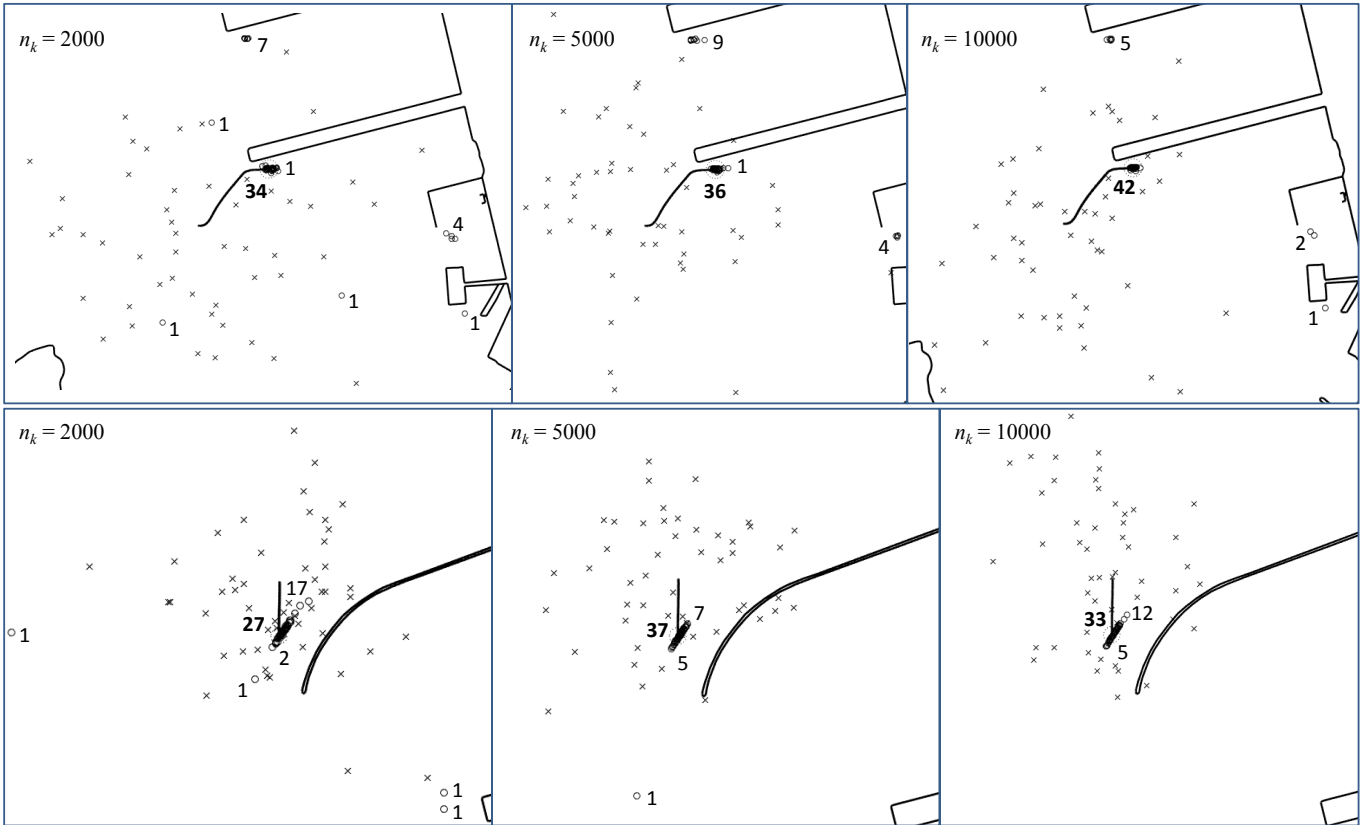


Fig. 11. Results of 50 MCL runs with the pier data (top row) and the breakwater data (bottom row). Each run started with n_k normally distributed particles (with CEP95 = 400 m) centered at a location shown with 'x', itself sampled from a normal distribution centered at the true initial location of the AUV (also with CEP95 = 400 m). The square meter with maximum particle density at the end of each run is marked by 'o'.

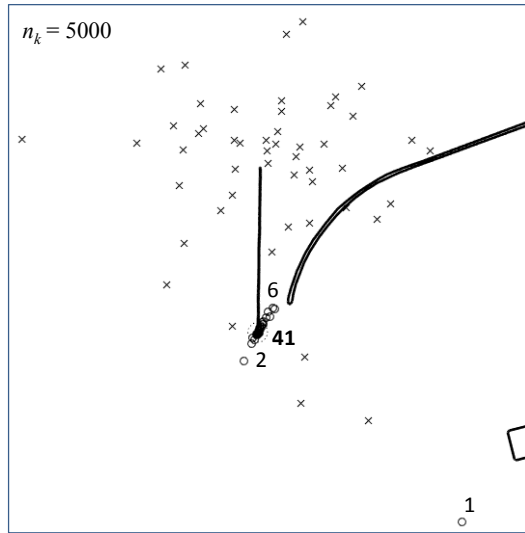


Fig. 12. Results of 50 MCL runs with the full breakwater data set. (Compare with corresponding panel in Fig. 11.)

and further investigation of problem-specific parameters, such as the choice of resampling interval and motion-model noise.

Improvements to the probabilistic measurement model is also an area of future work, including different ways of sampling the FLS data and the reference map, use of the 2-D map-matching method in Sec. II-D as an additional input to probabilistic localization, and further tuning of the model's parameters based on additional data collects.

An important aspect of in-water demonstration will be the integration of the UFIT module within the existing navigation system on the AUV. This integration may also lead to tighter coupling of MCL and FLS data processing.

The UFIT framework for localization presented in this paper is a promising and feasible addition to the existing set of tools for long-range precision navigation missions. By integrating new sources of information to provide accurate localization, it will extend the operational envelope of these missions.

ACKNOWLEDGMENT

This research was funded by the Office of Naval Research under contract N00014-12-C-0520. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

REFERENCES

- [1] R. N. Carpenter and M. R. Medeiros, "Concurrent mapping and localization and map matching on autonomous underwater vehicles," in *Oceans 2001. An Ocean Odyssey*, 5-8 Nov. 2001, pp. 380-389.
- [2] G. Peele, J. Adkison, J. De La Cruz, and S. Borkman, "Building a compact high-fidelity runtime database engine layer by layer," in *IMAGE 2011 Conference*, June 2011.
- [3] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131-149, 2014.
- [4] M. Walter, F. Hover, and J. Leonard, "Slam for ship hull inspection using exactly sparse extended information filters," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 1463-1470.
- [5] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater slam in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898-921, 2008.
- [6] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 4396-4403.
- [7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, ser. Intelligent robotics and autonomous agents. Cambridge and Mass: MIT Press, 2005.
- [8] G. Llor-Pujol, C. Sintes, and D. Gueriot, "Analysis of vernier interferometers for sonar bathymetry," in *OCEANS 2008*, pp. 1-5.
- [9] M. A. Masnadi-Shirazi, C. d. Moustier, P. Cervenka, and S. H. Zisk, "Differential phase estimation with the seamarcii bathymetric sidescan sonar system," *IEEE J. of Oceanic Eng.*, vol. 17, no. 3, pp. 239-251, 1992.
- [10] P. N. Denbigh, "Signal processing strategies for a bathymetric sidescan sonar," *IEEE J. of Oceanic Eng.*, vol. 19, no. 3, pp. 382-390, 1994.
- [11] G. Llor-Pujol, C. Sintes, and X. Lurton, "A new approach for fast and high-resolution interferometric bathymetry," in *OCEANS 2006 - Asia Pacific*, pp. 1-7.
- [12] X. Lurton and J.-M. Augustin, "A measurement quality factor for swath bathymetry sounders," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 4, pp. 852-862, 2010.
- [13] N. Allsop and Institution of Civil Engineers, *Breakwaters, Coastal Structures and Coastlines: Proc. of the Int. Conf. of the Institution of Civil Engineers, London, UK, Sept. 2001*. T. Telford, 2002.
- [14] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlking, C. Potthast, B. Zeisl, R. Rusu, S. Gedikli, and M. Vincze, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 80-91, 2012.
- [15] J. Bresenham, "Run length slice algorithm for incremental lines," in *Fundamental Algorithms for Comp. Graphics*, ser. NATO ASI Series, R. Earnshaw, Ed. Springer Berlin Heidelberg, 1991, vol. 17, pp. 59-104.
- [16] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: The Indian Journal of Statistics*, pp. 401-406, 1946.